UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/687,777	10/13/2000	Mukesh Patel	032481-021	1065

8791 7590 06/06/2005

BLAKELY SOKOLOFF TAYLOR & ZAFMAN
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1030

EXAMINER	
DAS, CHAMELI	

ART UNIT	PAPER NUMBER
2192	

DATE MAILED: 06/06/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	09/687,777	PATEL, MUKESH	
	Examiner	Art Unit	
	CHAMELI C. DAS	2192	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *09 May 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *43,50,52,54,56 and 109-128* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *43,50,52,54,56 and 109-128* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. This action is in response to the IDS filed on 5/9/05.

2. Claims 43, 50, 52, 56,109-128 are pending.

Double Patenting

3. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. See In re Goodman, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); In re Longi, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); In re Van Ornum, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); In re Vogel, 422 F.2d 438, 164 USPQ 619 (CCPA 1970);and, In re Thorington, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321© may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent is shown to be commonly owned with this application. See 37 CFR 1.130(b).

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

4. Instant claim 43, 50, 52, 54, 56, 109-128 are rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-41 of US Patent Number 6,826,749.

Although the conflicting claims are not identical, but they are not patentably distinct from each other because they are obvious variation of each other.

5. Claims 43, 50, 52, 54, 56, 109-128 are provisionally rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over

claims 1-59 of copending Application No. 11/062,012. Although the conflicting claims are not identical, they are not patentably distinct from each other because these two applications have a system comprising a central processing unit and a hardware accelerator to process stack-based instruction in cooperation with the CPU.

This is a provisional obviousness-type double patenting rejection because the conflicting claims have not in fact been patented.

Claim Rejections - 35 USC § 112

6. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

7. Claims 50, 52, 54, 126 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

In claim 50, at lines 6-7, the limitation "the immediate branch offset". There is insufficient antecedent basis for this limitation in the claim. This limitation is interpreted as "an immediate branch offset"

In claim 50, at line 8, the limitation "the current byte code instruction". There is insufficient antecedent basis for this limitation in the claim. This limitation is interpreted as "a current byte code instruction".

In claim 52, at lines 6-7, the limitation "the immediate branch offset". There is insufficient antecedent basis for this limitation in the claim. This limitation is interpreted as "an immediate branch offset".

In claim 52, at lines 7-8, the limitation "the virtual machine PC". There is insufficient antecedent basis for this limitation in the claim. This limitation is interpreted as "the virtual machine program counter".

In claim 52, at line 8, the limitation "the current byte code instruction". There is insufficient antecedent basis for this limitation in the claim. This limitation is interpreted as "a current byte code instruction".

In claim 54, at line 7, the limitation "the immediate field". There is insufficient antecedent basis for this limitation in the claim. This limitation is interpreted as "an immediate field".

In claim 54, at line 7, the limitation "register-based instruction being composed based the stack-based instruction" is not clear. This limitation is interpreted as "register-based instruction being composed based on the stack-based instructions".

In claim 126, at line 12, the limitation "the immediate branch offset". There is insufficient antecedent basis for this limitation in the claim. This limitation is interpreted as "an immediate branch offset".

In claim 126, at line 14, the limitation "the current byte code instruction". There is insufficient antecedent basis for this limitation in the claim. This limitation is interpreted as "a current byte code instruction".

In claim 126, at line 17, the limitation "the virtual machine PC". There is insufficient antecedent basis for this limitation in the claim. This limitation is interpreted as "the virtual machine program counter".

In claim 126, at line 18, the limitation "the return virtual machine program counter". There is insufficient antecedent basis for this limitation in the claim. This limitation is interpreted as "a return virtual machine program counter".

In claim 126, at line 20, the limitation "the virtual machine SiPush and BiPush byte codes". There is insufficient antecedent basis for this limitation in the claim. This limitation is interpreted as "virtual machine SiPush and BiPush byte codes".

In claim 126, at line 21, the limitation "the immediate field". There is insufficient antecedent basis for this limitation in the claim. This limitation is interpreted as "an immediate field".

Claim Objections

8. Claims 52 and 126 are objected to because of the following informalities:

In claim 52 in line 6 the limitation "JSR" should be replaced by Jump sub routine (JSR). Appropriate correction is required.

Similarly, claim 126, at line 15 the limitation "JSR" should be replaced by Jump sub routine (JSR). Appropriate correction is required.

Claim 126, line 2, the limitation "CPU" should be replaced by CPU core

Claim 126, at line 10, at the end of the line [[,]] should be replaced by ;

Claim 126, at line 14, at the end of the line [[,]] should be replaced by ;

Claim 126, at line 19, at the end of the line [[,]] should be replaced by ;

Claim 126, at line 22, at the end of the line [[,]] should be replaced by ;

Claim 126, at line 24, at the end of the line before "and [[,]] should be replaced by ;

Claim Rejections - 35 USC § 102

9. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371(c) of this title before the invention thereof by the applicant for patent.

The changes made to 35 U.S.C. 102(e) by the American Inventors Protection Act of 1999 (AIPA) and the Intellectual Property and High Technology Technical Amendments Act of 2002 do not apply when the reference is a U.S. patent resulting directly or indirectly from an international application filed before November 29, 2000. Therefore, the prior art date of the reference is determined under 35 U.S.C. 102(e) prior to the amendment by the AIPA (pre-AIPA 35 U.S.C. 102(e)).

Claims 43,122, are rejected under 35 U.S.C. 102(e) as being anticipated by "Designing Hardware to Interpret Virtual Machine Instructions" written by Otto Steinbusch,(Steinbusch), published on February, 1998 .

As per claim 43, Steinbusch discloses:

- a central processing unit (Steinbusch, page 13, section 3.1 in Introduction)

- a register file associated with the CPU core (Steinbusch, page 14, section "The register stack"), the register set is the register file (Steinbusch, page 33, section 4.3.1) and (page 18, "the instruction ***FIFO, which holds the CPU instructions*** that were generated by the VMI core. The instruction ***FIFO is a*** part of the ***register file***")
- a hardware accelerator to process stack-based instructions in cooperation with the CPU (Steinbusch, page 13, section 3.1 "INTRODUCTION", "PRLE set out to design a hardware module than can assist a CPU in executing Java Programs ... THE VMI core is placed between the CPU and the memory"), where "Java" instructions are stack-based instruction (Steinbusch , page 14), and "VMI" is the hardware accelerator, (page 31, section 4.1, "***VMI accelerate*** Java application" page 20, last paragraph), clearly indicates that VMI is hardware accelerator, hardware accelerator process stack-based instruction with the CPU core is shown in (page 16, in "The VMI range principal")
- the hardware accelerator marks variables associated with the stack-based instructions in the register file as modified when the variables are updated as a result of the processing of the stack-based instructions to enable selective writing of the variables as marked as modified to memory (page 18, lines 13-18), where VMI generates MIPS instructions that cause to update registers in the VMI, indicates that VMI can modify the registers by selective writing. On paged 32, 36 show that different kinds of registers and the registers hold

variables, page 37, 4th paragraph shows the registers are labeled (marked) indicates that the variables can be marked.

As per claim 122, Steinbusch discloses:

- a central processing unit (Steinbusch, page 13, section 3.1 in Introduction)
- a register file associated with the CPU core (Steinbusch, page 14, section "The register stack"), the register set is the register file (Steinbusch, page 33, section 4.3.1) and (page 18, "the instruction ***FIFO, which holds the CPU instructions*** that were generated by the VMI core. The instruction ***FIFO is a*** part of the ***register file***")
- a hardware accelerator to process stack-based instructions in cooperation with the CPU (Steinbusch, page 13, section 3.1 "INTRODUCTION", "PRLE set out to design a hardware module than can assist a CPU in executing Java Programs ... THE VMI core is placed between the CPU and the memory"), where "Java" instructions are stack-based instruction (Steinbusch , page 14), and "VMI" is the hardware accelerator, (page 31, section 4.1, "***VMI accelerate*** Java application" page 20, last paragraph), clearly indicates that VMI is hardware accelerator.

Wherein hardware accelerator maintains operand and variables required for processing the stack-based instructions in the register file (Steinbusch, page 14, page 49, last paragraph, page 50, section 5.4).

For claim 123, (page 10, section 6)

For claim 124, (Steinbusch, page 13, section 3.1).

For claim 125 (Steinbusch, page 13, section 3.1).

Claim Rejections - 35 USC § 103

10. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 50, 52, 54, 56, 109-121, 126-128 are rejected under 35 U.S.C. 103(a) as being unpatentable over Steinbusch and further in view of Tremblay, US 6,125,439

As per claim 50, Steinbusch discloses:

- a central processing unit (Steinbusch, page 13, section 3.1 in Introduction)
- a register file associated with the CPU core (Steinbusch, page 14, section "The register stack"), the register set is the register file (Steinbusch, page 33, section 4.3.1) and (page 18, "the instruction ***FIFO, which holds the CPU instructions*** that were generated by the VMI core. The instruction ***FIFO is a*** part of the ***register file***")
- a hardware accelerator to process stack-based instructions in cooperation with the CPU (Steinbusch, page 13, section 3.1 "INTRODUCTION", "PRLE set out to design a hardware module than can assist a CPU in executing Java Programs ... THE VMI core is placed between the CPU and the memory"), where "Java" instructions are stack-based instruction (Steinbusch , page 14),

and "VMI" is the hardware accelerator, (page 31, section 4.1, "***VMI accelerate*** Java application" page 20, last paragraph), clearly indicates that VMI is hardware accelerator

- hardware accelerator ... byte code instruction (page 32, where "byte code counter" is the program counter, and "sign extending" operations are "overflow/underflow" operation, page 41, 3rd paragraph, page 42 , 4th paragraph, "conditional jump ... CPU to write back a value".
- Steinbusch discloses GOTO instruction (page 42, 2nd paragraph). Steinbusch does not specifically disclose GOTO_W instruction. However, Tremblay discloses GOTO_W instruction (col 71, lines 49). The modification would be obvious because one of the ordinary skill in the art would be motivated to implement the system to write back the value into the register.

As per claim 52, Steinbusch discloses:

- a central processing unit (Steinbusch, page 13, section 3.1 in Introduction)
- a register file associated with the CPU core (Steinbusch, page 14, section "The register stack"), the register set is the register file (Steinbusch, page 33, section 4.3.1) and (page 18, "the instruction ***FIFO, which holds the CPU instructions*** that were generated by the VMI core. The instruction ***FIFO is a*** part of the ***register file***")
- a hardware accelerator to process stack-based instructions in cooperation with the CPU (Steinbusch, page 13, section 3.1 "INTRODUCTION", "PRLE

set out to design a hardware module than can assist a CPU in executing Java Programs ... THE VMI core is placed between the CPU and the memory"), where "Java" instructions are stack-based instruction (Steinbusch , page 14), and "VMI" is the hardware accelerator, (page 31, section 4.1, "***VMI accelerate*** Java application" page 20, last paragraph), clearly indicates that VMI is hardware accelerator.

- Steinbusch discloses conditional jump and unconditional jump instructions (page 32, page 42, 3rd paragraph). Steinbusch does not specifically disclose JSR_W.

However, Tremblay discloses JSR_W instruction (col 71, lines 65-67 and col 72 lines 5-15). The modification would be obvious because one of the ordinary skill in the art would be motivated to implement the system to write back the value into the register.

As per claim 54, Steinbusch discloses:

- a central processing unit (Steinbusch, page 13, section 3.1 in Introduction)
- a register file associated with the CPU core (Steinbusch, page 14, section "The register stack"), the register set is the register file (Steinbusch, page 33, section 4.3.1) and (page 18, "the instruction ***FIFO, which holds the CPU instructions*** that were generated by the VMI core. The instruction ***FIFO is a*** part of the ***register file***")

- a hardware accelerator to process stack-based instructions in cooperation with the CPU (Steinbusch, page 13, section 3.1 "INTRODUCTION", "PRLE set out to design a hardware module than can assist a CPU in executing Java Programs ... THE VMI core is placed between the CPU and the memory"), where "Java" instructions are stack-based instruction (Steinbusch , page 14), and "VMI" is the hardware accelerator, (page 31, section 4.1, "***VMI accelerate*** Java application" page 20, last paragraph), clearly indicates that VMI is hardware accelerator.

Steinbusch discloses push onto the operand stack (page 31, section 4.2, 5th paragraph). Steinbusch does not specifically disclose "Sipush" and "Bipush" byte codes. However, Tremblay discloses "Sipush" and "Bipush" (col 43 lines 8-20) and sign extension (col 65 lines 54-55). The modification would be obvious because one of the ordinary skill in the art would be motivated to insert the new items into the stack.

For claim 56 see the rejection of claim 54 above.

For claim 109 (Steinbusch, page 13, section 3.1, page 31 and page 32).

For claim 110 (Steinbusch, page 13, section 3.1, page 31 and page 32).

For claim 111 (Steinbusch, page 13, section 3.1, page 31 and page 32).

For claim 112 (Steinbusch, page 13, section 3.1, page 31 and page 32).

For claim 113 (Steinbusch, page 13, section 3.1, page 31 and page 32).

For claim 114 (Steinbusch, page 13, section 3.1, page 31 and page 32).

For claim 115 (Steinbusch, page 13, section 3.1, page 31 and page 32).

For claim 116 (Steinbusch, page 13, section 3.1, page 31 and page 32).

For claim 117 (Steinbusch, page 13, section 3.1, page 31 and page 32).

For claim 118 (Steinbusch, page 13, section 3.1, page 31 and page 32).

As per claim 119, Steinbusch discloses:

- a central processing unit (Steinbusch, page 13, section 3.1 in Introduction)
- a register file associated with the CPU core (Steinbusch, page 14, section "The register stack"), the register set is the register file (Steinbusch, page 33, section 4.3.1) and (page 18, "the instruction ***FIFO, which holds the CPU instructions*** that were generated by the VMI core. The instruction ***FIFO is a*** part of the ***register file***")
- a hardware accelerator to process stack-based instructions in cooperation with the CPU (Steinbusch, page 13, section 3.1 "INTRODUCTION", "PRLE set out to design a hardware module than can assist a CPU in executing Java Programs ... THE VMI core is placed between the CPU and the memory"), where "Java" instructions are stack-based instruction (Steinbusch , page 14), and "VMI" is the hardware accelerator, (page 31, section 4.1, "***VMI accelerate*** Java application" page 20, last paragraph), clearly indicates that VMI is hardware accelerator.

Tremblay discloses the hardware accelerator maintains an operand stack for the stack-based instructions in the register file (Tremblay, col 17 lines 1-17), loads variables ... register file (col 27 lines 47-49), stack cache is users as register file is shown in (col 17

lines 11-15), operand stack in the register file define a ring buffer in conjunction with an overflow/underflow ... memory (Tremblay, col 18 lines 48-57). The modification would be obvious because the ring buffer ensures that the stack grows and shrinks in a predictable manner to avoid overflows or overwrites.

For claim 120 (Steinbusch, page 13, section 3.1, page 31 and page 32).

For claim 121 (Steinbusch, page 13, section 3.1).

For claim 126, (see the rejection of claims 43, 50, 52, 54, 56, 119 and 122).

For claim 127 (Steinbusch, page 13, section 3.1, page 31 and page 32).

For claim 128 (Steinbusch, page 13, section 3.1, page 31 and page 32).

11. The prior art made or record and not relied upon is considered pertinent to applicant's disclosure.

TITLE: Bytecode program interpreter apparatus and method with pre-verification of data type restrictions and object initialization, US 6477702 B1

TITLE: Cache circuit with programmable sizing and method of operation, US 5940858 A

TITLE: Process of executing a method on a stack-based processor, US 6125439 A

TITLE: Shadow translation look-aside buffer and method of operation, US 5946718 A

12. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chameli Das whose telephone number is 571-272-2696.

The examiner can normally be reached on Monday-Friday from 7:00 A.M. to 3:30 P.M.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor Tuan Dam can be reached at 571-272-2695. The fax number for this group is (703) 872-9306.

An inquiry of general nature or relating to the status of this application or proceeding should be directed to the group receptionist whose telephone number is 571-272-2100.

Chameli C. Das

CHAMELI C. DAS
PRIMARY EXAMINER

5/28/05